

BEST AVAILABLE COPY



Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM),
European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE,
ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, SK,
TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ,
GW, ML, MR, NE, SN, TD, TG).

— *before the expiration of the time limit for amending the
claims and to be republished in the event of receipt of
amendments*

Published:

— *with international search report*

*For two-letter codes and other abbreviations, refer to the "Guid-
ance Notes on Codes and Abbreviations" appearing at the begin-
ning of each regular issue of the PCT Gazette.*

WIRELESS DEVICE ATTACHMENT AND DETACHMENT SYSTEM, APPARATUS AND METHOD

BACKGROUND

Related Applications

This application claims the benefit of co-pending U.S. Provisional Application Serial No. 60/314,524 filed August 22, 2001.

Field of the Invention

Aspects of the present invention relate in general to the field of wireless communications.

Background

One common way of attaching peripherals to a computer is via a Universal Serial Bus (USB) interface. One of the most convenient features of USB is the ability to add and remove devices from the bus during normal operation ("hot-plugging"). To achieve this "hot-plugging" feature, the host periodically polls each attached USB hub device to determine the state of each of the hub's active downstream ports. If the hub detects that a device is attached to a port, it enables the port and then enumerates the device. If the hub detects that a previously attached device is detached, the port is disabled. In a wired environment, the hub determines the attachment and detachment of devices by monitoring the differential data lines within the USB cable to detect changes in voltage levels. The hub then sets bits in its status register indicating any changes to its ports. The host subsequently reads the status register on the next poll period. The process of detecting device attachment and detachment by monitoring differential data lines is clearly not feasible in a wireless environment.

Due to the ease of installation and use by consumers, USB interfaces have grown in popularity and are included on many computer peripherals. Despite its benefits, USB suffers from the multitude of cables required to attach peripherals.

BRIEF DESCRIPTION OF THE DRAWINGS

5 FIGS. 1A and 1B illustrate embodiments of a system to attach and detach wireless peripheral devices to a host in a wireless system.

FIG. 2 is a block diagram of a wireless module to attach and detach wireless peripheral devices to a host in a wireless system.

FIG. 3 is a block diagram of a media access control (MAC) layer to attach and
10 detach wireless peripheral devices to a host in a wireless system.

FIG. 4 is a block diagram of an embodiment of a protocol management interface (PMI) to attach and detach wireless peripheral devices.

FIG. 5 is a block diagram of a poll message packet data structure, to attach and detach wireless peripheral devices to a host in a wireless system.

15 FIG. 6 is a block diagram of a response message packet data structure, to attach and detach wireless peripheral devices to a host in a wireless system.

FIGS. 7A-7B are a flowchart of a host method 700, to attach and detach wireless peripheral devices to a host in a wireless system.

FIG. 8 is a flowchart of a peripheral method 800, to attach and detach wireless
20 peripheral devices.

FIG. 9 depicts a hub status change register to attach and detach wireless peripheral devices.

FIG. 10 depicts an example of a collision avoidance scheme to attach and detach wireless peripheral devices.

FIGS. 11A-11B are charts depicting possible state corresponding to ports, to attach and detach wireless peripheral devices to a host in a wireless system.

FIG. 12 is an exemplary diagram of port information stored in a memory structure, to attach and detach wireless peripheral devices to a host in a wireless system.

5 FIGS. 13A-13B are a flowchart of an alternate host method 1300, to attach and detach wireless peripheral devices to a host in a wireless system.

FIG. 14 is a flowchart of an alternate host method 1400, to attach and detach wireless peripheral devices to a host in a wireless system.

10 FIG. 15 is a flowchart of a process response method 1500, to attach and detach wireless peripheral devices to a host in a wireless system.

FIGS. 16A and 16B depict memory tables to track the attachment and detachment of wireless peripheral devices to a host in a wireless system.

DETAILED DESCRIPTION

Aspects of the present invention include a system, apparatus and method to attach
15 and detach wireless peripheral devices. In some aspects, wireless networks enable embodiments that are fully compatible with the Universal Serial Bus. Some embodiments enable a wireless host device to detect the attachment of a new wireless peripheral, to detect detachment of a previously attached wireless peripheral, and to monitor continued presence of a previously attached device.

20 The embodiments of the invention may be independent of any underlying wireless communication technology. It is understood that although the following embodiments are disclosed using a wireless connection to implement a wireless communications framework compatible with Universal Serial Bus, other embodiments using other wireless communications protocols and technology are possible. Examples of alternative wireless

communication protocols and technology compatible with embodiments of the present invention include, but are not limited to: ultra-wide-band (UWB), a wireless implementation of Institute of Electrical and Electronics Engineers (IEEE) Standard No. 1394 ('wireless Firewire'), Institute of Electrical and Electronics Engineers (IEEE) Standard Nos. 802.11a, IEEE 802.11b ("Wi-Fi"), IEEE 802.11g, IEEE 802.15 (WPAN), Bluetooth, and a wireless implementation of the RS-232 protocol. For convenience, embodiments below will now be discussed using the USB protocol.

FIG. 1A is a simplified diagram depicting system 100, constructed and operative in accordance with an embodiment of the present invention. System 100 is configured to attach and detach wireless peripheral devices 20 to a wireless host device 10 in a wireless system.

In some embodiments, wireless peripheral devices 20 are expected to be limited in resources, lacking a man-machine interface for network-related tasks. These network-related tasks include entering peripheral device encryption keys. Also, in many embodiments, wireless peripheral devices 20 may not have a processor available to perform complex communication tasks.

As shown in FIG. 1B, constructed and operative in accordance with an embodiment of the present invention, the wireless devices 10 and 20 depicted in system 100 may comprise host 15 and peripheral devices 25 coupled to wireless modules 110. For example, wireless host device 10 may be further comprised of host 15 and a wireless module 110x.

Wireless modules 110a-x are any devices that allow a conventional computing device to communicate with other wireless modules 110 or wireless devices 10, 20 using a method embodiment of the present invention, as described in the claims below. The host device 15 interacts with the wireless system through wireless module 110x. The wireless

module 110x sends data received from the host 15 to wireless peripheral devices 20 and passes on data received from wireless peripheral devices 20 through the system to the host 15.

FIG. 2 is a block diagram of wireless module 110, constructed and operative in accordance with an embodiment of the present invention. It is well understood by those knowledgeable in the art that the elements of FIG. 2 may be implemented as structures in hardware, firmware, or as software instructions and data encoded on a computer-readable storage medium. Wireless module 110 comprises device interface 102, media access control layer (MAC) 104 and physical layer (PHY) 106. In some embodiments, wireless module 110 may also include an antenna (not shown), as is known in the art.

Device interface 102 is any interface, as is known in the art, which allows wireless module 110 to interface with host 15 or peripheral device 25. In some embodiments, device interface 102 may emulate a standard communications protocol so that host 15 or peripheral device 25 does not realize that it is communicating with another wireless device. For example, in wireless USB embodiments, device interface 102 may emulate a USB interface.

Media access control layer 104 is any structure that governs access to the wireless transmission media. As shown in FIG. 2, media access control layer 104 is coupled to device interface 102 and physical layer 106, enabling device interface 102 to exchange data with the physical layer 106.

Physical layer 106 is any structure that provides the procedures involved in transferring a single bit over the wireless medium being used by the wireless module 110.

FIG. 3 is a block diagram of a media access control layer 104, constructed and operative in accordance with an embodiment of the present invention. In this

embodiment, media access control layer 104 comprises a protocol management interface 300 and a key management and encryption unit (KMEU) 310.

Protocol management interface 300 is the structure that enables media access control layer 104 to communicate using a particular wireless communications protocol, as described above. For example, in a wireless Universal Serial Bus (wireless USB) embodiment, protocol management interface 300 processes messages from device interface 102 and regulates the message access to the physical layer 106 using the Universal Serial Bus protocol.

In a conventional (wired) USB system, a USB port is a physical structure, and a USB host device periodically polls each attached hub device to determine the state of each of the hub's active downstream ports. In a wireless USB embodiment, each port is no longer a physical structure, but a virtual structure to be connected via physical layer 106. In the wireless embodiment, protocol management interface 300 may be the structure that monitors the downstream (virtual) USB ports. In embodiments using different wireless protocols, a corresponding protocol may be implemented in protocol management interface 300.

Key management and encryption unit 310 performs encryption and decryption of messages. As shown in FIG. 3, KMEU 310 may be situated in the MAC layer 104. In some embodiments, however, media access control layer 104 may only comprise protocol management interface 300. In such embodiments, the key management and encryption unit 310 may be located in, or functionally distributed within the wireless module 110, among other parts of the path between devices 15 or 25 and the antenna, or may be omitted entirely. Therefore, the location of the key management and encryption unit 310 should not be viewed as limiting the scope of the embodiments of the present invention, which are limited solely by the claims.

FIG. 4 is a block diagram of an alternate embodiment of media access control layer 104, constructed and operative in accordance with an embodiment of the present invention. In this embodiment, media access control layer 104 additionally comprises a computer readable medium 400.

5 Computer readable medium 400 may be any computer-readable memory known in the art. This definition encompasses, but is not limited to: Read Only Memory (ROM), Random Access Memory (RAM), flash memory, Erasable-Programmable Read Only Memory (EPROM), non-volatile random access memory, memory-stick, magnetic disk drive, transistor-based memory or other computer-readable memory devices as is known
10 in the art for storing and retrieving data. An example of such data is port data information 410.

FIGS. 7A-7B are a flowchart of a host process 700, an embodiment to attach and detach a wireless peripheral device 20 to a wireless host device 10 in a wireless system 100, constructed and operative in accordance with an embodiment of the present
15 invention.

To implement the "hot-plugging" feature of wired USB in a wireless scenario, host process 700 uses a polling scheme in which messages are passed over the wireless link, between the host device 10 and the peripheral device 20, in order to determine the state of each port supported by the host device 10. FIGS. 5 and 6 show an exemplary
20 format of these messages, constructed and operative in accordance with an embodiment of the present invention. A poll message 500 comprises a poll message header 502 and a peripheral device address 504. If a peripheral is attached to the port, the peripheral device address 504 contains the unique address of the peripheral device. Otherwise, the peripheral device address 504 contains the "No Device" address. A response message
25 600 comprises a response message header 602 and a unique peripheral device address

604. During each polling period, a specific poll message 500 is generated by the wireless module 110x for each of the ports that need to be polled. After sending out a poll message 500 through a port, the wireless module 110x awaits a response message 600 from the wireless module 100a-n. Based on the response message 600 received, appropriate action is performed by the wireless module 110x as described below. An embodiment of the various port states and actions performed are shown in FIGS. 11A and 11B, constructed and operative in accordance with an embodiment of the present invention. These states are described in greater detail below.

To retain the state of all downstream ports, the host maintains an internal data structure. This data structure may be any information structure known in the art capable of containing an entry for each port available to the host, including, but not limited to: arrays, linked lists, trees, or database schemas, and the like.

An exemplary port data structure 1200 is shown in FIG. 12, constructed and operative in accordance with an embodiment of the present invention. As shown in FIG. 12, each port 1208A-D is associated with three elements, a port identifier field 1202, an address ("Addr") field 1204, and a retries ("Retries") field 1206.

It is understood, by those of ordinary skill in the art, that the number of ports 1208 shown is for illustrative purposes only, and that the number of ports is not limited by the process 700, but that some embodiments may be limited by their individual implementation. For example, the Universal Serial Bus v. 2.0 specifications limit the number of ports to 127. Consequently, Universal Serial Bus v. 2.0 embodiments may be similarly limited.

Returning to FIG. 12, the address field 1204 corresponds to either a unique peripheral address assigned to the port 1208 indicating an attached device, or a predetermined unique "No Device" address (i.e. the detached state). Note that the unique

peripheral address is different from a USB address assigned by USB host software. The retries field 1206 indicates the number of times the host will retransmit a poll message to a peripheral address, in the case where the device fails to respond to its own address.

This encompasses situations where either the polling message or the device response is
5 lost during transmission. Upon power-up, the address field 1204 for each port 1208 is set to the "No Device" state, while the retries field 1206 is set to some initial predetermined value. FIG. 12 depicts an example of a host with four ports 1208A-D. Ports 1208B-C are occupied. Port 1208C has been polled three times with no response.

Returning to FIG. 7A, initially, at block 702, the wireless hub module 110x
10 initializes the invitation count to a predefined number. At decision block 704, if the port is currently marked as "free" or "not in use," flow continues at decision block 706. If the port is currently in use, flow continues at block 708.

At decision block 706, if the invitation count is greater than zero, flow continues at block 708. If the invitation count is not greater than zero, no new invitations need to be
15 sent. The hub checks to see if all the ports have been polled at decision block 722. If all the ports have not been polled, then the hub moves onto the next port at block 724, and flow returns to block 704. If all the ports have been polled, as determined at block 722, process 700 ends.

At block 708, the hub sends out a poll message, and awaits a response from the
20 device for a specified amount of time at block 710. If a timeout occurs as determined at block 712, it may either mean that the peripheral is no longer within range, or that the poll message or response message was lost during transmission. The flow then continues at block 726 in FIG. 7B.

If a device is still within range of the hub and functioning, it sends a response
25 message to the hub, with the address field 1204 set to its unique peripheral address. If the

hub receives a response before the timeout period as determined at block 712, it checks to make sure that the received response is valid at decision block 714. If the received response is invalid, flow continues at block 726 in FIG. 7B. If a valid response is received, the hub checks to see if the port is currently marked free at decision block 716.

5 If the port is currently marked free, it indicates that the response was received from a newly attaching device. The hub records the address 1204 of the new device and initializes the retries field 1206 of the concerned port to MaxRetries at block 720. Obviously, embodiments that do not use unique peripheral addresses do not record the address at block 720. The hub then checks to see if all the ports have been polled at
10 decision block 722.

If the hub determines that the port is currently in use at block 716, it signifies that the device is still active. The hub resets the retries field 1206 of that port to MaxRetries at block 718. Then, as in the other case, the hub checks to see if all the ports have been polled at decision block 722. If all the ports have not been polled, then the hub moves
15 onto the next port at block 724, and flow returns to block 704. If all the ports have been polled, as determined at block 722, process 700 ends.

Continuing onto FIG. 7B, if the port is currently marked free as determined at block 726, the hub decrements the invitation count, and if all the ports have not been polled as determined at decision block 722, the hub moves onto the next port at block
20 724, and flow returns to block 704. If all the ports have been polled, process 700 ends. Alternatively, if the port is currently in use as determined by the hub at decision block 726, then the hub decrements the current port's retries field 1206 at block 730, and checks to see if the retries field equates to zero at decision block 732. If the retries field 1206 is zero, it means that the device is now out of range. The hub detaches the device and
25 notifies host 15 of the change, at block 734. Then, the hub checks to see if there are any

more ports to be polled at decision block 722. If the retries field 1206 is not zero as determined by decision block 732, the hub does not take any further action. The flow moves on to decision block 722 as before, to determine if there are any more ports to be polled. If there are more ports to be polled as determined at decision block 722, the hub
5 moves to the next port at block 724 and flow returns to block 704. If all the ports have been polled, process 700 ends.

Due to the potential large number of ports, many of which may be unused, numerous unnecessary invitations for newly joining devices could be sent by the host 15. If no response is received to the previous "No Device" polling invitation, it may mean
10 that there is no new device wanting to attach, or that either the invitation or response to the invitation was garbled during transmission. Process 700 accounts for this by maintaining a count of invitation messages sent. If there is no response to a "No Device" polling invitation, the invitation count is decremented, at block 728. Then, the hub checks to see if the invitation count has reached zero, at decision block 706. If the count reaches
15 zero, the host stops sending further invitations during this polling cycle. The count is reinitialized at block 702 during the next round of polling.

A situation may arise where two (or more) devices attempt to connect to the system at the same time. This may happen, for instance, upon power-up, when a number of devices are already within range of the host, and one or more of the devices
20 simultaneously respond to a host invitation. To handle this case, a technique based on time slots may be applied. Each device receiving an invitation to the system waits a random number of time slots (0..n). During the waiting period, it senses the channel to determine whether any other device is transmitting. If so, the device backs off and waits for another invitation from the host. This channel access method probabilistically reduces
25 the number of collisions that can occur during the attachment phase.

An embodiment of a host method 700, implemented in a wireless host module 110x, is shown below. During normal polling by the host computer, the hub cycles through each entry in the data structure depicted in FIG. 12, sending out a poll message with the stored address for each port.

5

```

1  sendInvitation = MaxInvitations;      // invite new devices
2  for(port=0; port<MaxPorts; port++) { // for all ports...
3      if ((addr[port] != NoDevice) || (sendInvitation > 0)) {
4          SendPoll(addr[port]);          // send polling message
10     ReceiveResp(&a); // wait for a response
           // (a = NoDevice if no response received)
6         if (addr[port] == NoDevice) { // port currently free
7             if (a != NoDevice) {      // new device attached
8                 addr[port] = a;        // remember its address
15             retries[port] = MaxRetries;
           status[port] = attached; // notify host of change
11         } else { // No response received
12             // Decrement invitation count
13             sendInvitation = sendInvitation - 1;
20         }
14     } else {                          // port in use
15         if (a == addr[port])          // device still alive
16             retries[port] = MaxRetries;
17         else { // Invalid response or no response from device
25         if (retries[port] == 0) { // retry threshold exceeded
19             addr[port] = NoDevice; // device detached
20             status[port] = detached; // notify host of change
21         } else {
22             // Allow more attempts
30         retries[port] = retries[port] - 1;
23     }
24     }
25     }
26     }
27     }
28 }

```

35

Host Method Pseudo-Code Embodiment

FIG. 8 depicts a flowchart of a peripheral method 800, an embodiment to attach and detach a wireless peripheral device 20 to a wireless host device 10 in a wireless system 100, constructed and operative in accordance with an embodiment of the present invention. Within peripheral method 800, wireless peripheral device 20 waits to receive a polling message, block 802. After a polling message is received, wireless peripheral device 20 responds to the polling message, block 804. Embodiments that use a unique

peripheral address respond with their unique peripheral address; embodiments that do not use a unique peripheral address may respond with the received port number.

FIG. 9 depicts a hub status register 900 to attach and detach wireless peripheral devices to a host in a wireless system, constructed and operative in accordance with an embodiment of the present invention. The host device 10 polls the hub status register 900 periodically, and performs the appropriate action according to USB specifications.

FIG. 10 depicts an example of a collision avoidance scheme to attach and detach wireless peripheral devices to a host in a wireless system, constructed and operative in accordance with an embodiment of the present invention. In the example diagram, two peripheral devices 20a and 20b attempt to respond to an invitation message by the host device 10. Assume that the peripheral device A (20a) randomly chooses to wait one time slot while peripheral device B (20b) chooses to wait four time slots before transmitting a response. In time slot one, peripheral device B senses that another peripheral device, peripheral device A, has started transmitting a response and accordingly backs off and waits for the next invitation message to arrive. Note that collisions can occur only when newly arriving devices try to connect to the wireless hub. Already attached devices are polled directly and do not need to compete for the channel. Consequently, there is no need to wait between the polling message from the host device 10 and the response from the attached peripheral device 20.

FIGS. 13A-13B are a flowchart of a host process 1300, an alternate embodiment to attach and detach a wireless peripheral device 20 to a wireless host device 10 in a wireless network 100, constructed and operative in accordance with an embodiment of the present invention.

To implement this “hot-plugging” feature for wireless USB, host process 1300 uses a polling scheme in which messages are passed over the wireless link, between the

host device 10 and the peripheral device 20, in order to determine the state of each port supported by the host device 10. As in other embodiments, process 1300 retains the state of all downstream ports, the host maintains an internal data structure. The data structure may be any information structure known in the art capable of containing an entry for each
5 port available to the host, including, but not limited to: arrays, linked lists, trees, or database schemas, and the like.

Initially, at block 1302, the wireless host device 10 sends a poll message through a port. If the port is marked as “free”, the address field 1204 of the “poll” message is set to the “No Device” address; this is an invitation for new devices to join the network.
10 Alternatively, if the port is marked as “in use”, the address field 1204 of the “poll” message is set to the unique peripheral address. At block 1304, the host waits to receive a response or times out.

At decision block 1306, if the port is currently marked as “free” or “not in use,” flow continues at block 1308. If the port is currently in use, flow continues at block 1318.

15 If there is no response from the polling, as determined by decision block 1308, the port remains marked as free, at block 1312. Alternatively, if a response is received at decision block 1308, flow continues at block 1310. Upon receiving this response, the host
15 marks that port 1208 as occupied by setting the address field 1204 to the received address, and by setting the retries field 1206 to the initial value. The host also sets the
20 appropriate bit in the host status register 900 informing the host computer of the newly attached device. Flow continues at block 1314.

If all the ports have not yet been polled, as determined by block 1314, process 1300 continues to the next message port, block 1316, and flow returns to block 1302.

If all the ports have been polled, as determined by block 1314, process 1300 ends.

Continuing on FIG. 13B, if the “poll” message corresponds to an attached port, as determined at block 1306, the host then waits for a return response from the device for a specified amount of time. If the device is still within range and functioning as determined by block 1318, it responds with a “response” message, with the address field 1204 set to
5 its unique peripheral address, which is the same address as specified in the “poll” message. The host continues communicating with the device at block 1320. The host records the port 1208 and unique peripheral address, and resets the retries field 1206 count to its initial value. If, within the timeout period, the host does not receive a response, at decision block 1322, it may mean that the device is no longer present (i.e.
10 detached), or that an error occurred in either the transmission of the “poll” message, or the response message. In this case, if the retries field 1206 count is greater than 0, the host eventually reattempts to make contact at block 1326. The host decrements the retries field 1206 count by one and proceeds to the next port. Flow returns to block 1302.

If the retries field 1206 count equals 0, this signals to the host that the device is
15 out of range (i.e. detached). The host detaches the device, informing the host computer of the detachment, block 1324. In some embodiments, the host resets the retries field 1206 count to the initial value, and sets the appropriate bit in a host status register, block 1324.

Due to the potential large number of ports, many of which may be unused, numerous unnecessary invitations could be sent by the host 15. The process 1300 may
20 account for this situation by only transmitting additional invitations if there was a response to the previous invitation. In some embodiments, this may be implemented through maintaining an invitation flag indicating whether the last invitation was accepted. If not, then no more invitation messages (i.e. “poll” messages), with the address field 1204 set to “No Device,” are sent during this polling period.

In the operation of some embodiments, invitations or responses to invitations are lost during transmission. The process 1300 may account for this by maintaining a count of invitation messages sent. If there is no response to the message, the count is decremented. If the count reaches zero, the host stops sending invitations. The count is
5 reinitialized whenever a response is received.

FIG. 14 illustrates a flowchart of a host polling period process 1400, another alternate embodiment to attach and detach a wireless peripheral device 20 to a wireless host device 10 in a wireless network 100, constructed and operative in accordance with an embodiment of the present invention. In this embodiment, neither a unique peripheral
10 address nor a predetermined unique "No Device" address is required. Instead, the wireless host device 10 maintains a table along with an index into a table. The size of the table is equal to the number of available USB ports supported by the wireless host device 10, and therefore is indexed by port number. Examples of such a table are shown in FIGS. 16A and 16B, constructed and operative in accordance with an embodiment of the present
15 invention. At each entry in the table, the wireless host device 10 stores a retry count for an associated port. A count of zero indicates that the port is available for new devices. At power-on, the retry count for each port is cleared indicating no devices are attached, as depicted in FIG. 16A.

During each polling period 1400, the wireless host device 10 in turn cycles
20 through a number of entries in the table. At decision block 1402, a determination is made on whether a device is attached to the port.

For each entry in the table, a poll message is generated, block 1408. The contents of the request are the port number (i.e. the current index), and a status bit indicating the current state of the port. For retry counts of zero, the status bit is set to indicate that the
25 port is available for new devices, otherwise it is set to indicate that the port is currently

occupied. The wireless host device 10 then sends out the message and waits for a “Response” message or times out, block 1410. If a timeout occurs, it may mean that there are no new devices wishing to attach. In this case, the wireless host device 10 simply increments the index and continues.

5 Any device wishing to attach, listens for the “poll” message which corresponds to an unoccupied port. Upon receiving such a message, the device stores the port number and responds by issuing a “response” message with the same port number as specified in the “poll” message.

 A response processing sub-task 1500 handles the “response” message, as
10 illustrated in FIG. 15, constructed and operative in accordance with an embodiment of the present invention. A determination is made on whether the port is currently marked as free (i.e. “not occupied”), block 1502. The wireless host device 10, upon receiving the response, sets the retry count to a predefined non-zero number, for the port to indicate that the port is now occupied, block 1504. Sub-task 1500 notifies the host computer of
15 the newly attached device, and increments the index at block 1504.

 At decision block 1506, a determination is made on whether the device attached to the port is still active. If so, communications are continued with the device, block 1508. If the device is not active, sub-task 1500 detaches the device and informs the host computer, block 1514. Some embodiments may attempt to reestablish communication with the
20 device, blocks 1510-1512, before detaching the device.

 If all the ports have not yet been polled, as determined by block 1412, process 1400 continues to the next message port, block 1414, and flow returns to block 1402.

 If all the ports have been polled, as determined by block 1412, process 1400 ends.

 The following pseudo-code represents one possible implementation of process
25 1400 and sub-task 1500.

```

1.   sendInvitation = MaxInvitations; // invite new devices
2.   for (port=0; port<MaxQueryPorts; port++)
3.       if ((retries[port] > 1) {
4.           request = AttachedRequest; // device attached to port
5   5.   } else if (sendInvitation > 0) {
6.       request = InviteRequest; // invite new attachment
7.   } else continue; // else continue with next port
8.   SendPoll(port, request); // send a polling message
9.   ReceiveResp(&resp); // wait for a response
10  10.  process_response(port, resp); // process the response
11.  }
12.  // response processing task
13.  process_response(port, resp) {
14.      if (retries[port] == 0) { // port currently free
15  15.      if (resp == port) { // new device attached
16.          retries[port] = MaxRetries; // remember its address
17.          status[port] = attached; // notify host of change
18.      } else if (resp == NoResponse) {
19.          // Decrement invitation count
20  20.          sendInvitation = sendInvitation - 1;;
21.      }
22.      } else { // port in use
23.          if (resp == port) { // device still alive
24.              retries[port] = MaxRetries;
25  25.          } else { // Invalid response or no response from device
26.              if (retries[port] == 0) { // retry threshold exceeded
27.                  status[port] = detached; // notify host of change
28.              } else { // allow more attempts
29.                  retries[port] = retries[port] - 1;
30  30.              }
31.          }
32.      }
33.  }

```

Alternate Embodiment Host Method Pseudo-Code

35 As described above, the embodiments of the present invention may include a
single host and multiple peripheral devices. It is understood that in some embodiments
other hosts and other peripheral devices may be collocated. Collocation may bring about
interference concerns. Such concerns include garbled or lost communications, thereby
impairing the ability of peripherals to attach to the host, or causing the host to perform
40 unwanted detachments of peripherals. Several techniques that provide channelization
solutions, as is known in the art, include Time Division Multiple Access (TDMA), Carrier
Sensing Multiple Access (CSMA), and CDMA (Code Division Multiple Access).

It is further understood that in such embodiments, collocation of multiple hosts and devices provides enables the sharing of devices between hosts. There are numerous approaches that permit device sharing. For example, mechanical devices such as knobs and buttons may allow manual channel selection for hosts and peripherals. Alternatively, 5 host software may provide a user interface allowing peripheral devices to select a desired host. Additionally, dynamic channel selection may be implemented using ranging techniques as is known in the art.

The previous description of the embodiments is provided to enable any person skilled in the art to practice the invention. The various modifications to these 10 embodiments will be readily apparent to those skilled in the art, and the generic principles defined herein may be applied to other embodiments without the use of inventive faculty. Thus, the present invention is not intended to be limited to the embodiments shown herein, but is to be accorded the widest scope consistent with the principles and novel features disclosed herein.

15 WHAT IS CLAIMED IS:

CLAIMS

- 1 1. A host apparatus comprising:
2 a downstream port configured to receive a response from a wireless peripheral
3 device; and
4 a control layer configured to poll the downstream port, to determine the state of
5 the downstream port;
6 a computer readable memory configured to store updated port information with
7 the state of the downstream port from the control layer.
- 1 2. The apparatus of claim 1, further comprising:
2 a status register configured to store a state of the downstream port.
- 1 3. The apparatus of claim 2, wherein the control layer is further configured to
2 poll the downstream port by sending a polling message, the polling
3 message comprising a message header identifier.
- 4 4 The Apparatus of claim 3, wherein the control layer is further configured
5 to re-poll the downstream port when a response is not received from the
6 wireless peripheral device on the downstream port.
- 1 5. The apparatus of claim 4 wherein the control layer is further configured to
2 poll the downstream port until a retry count is exhausted or a response is
3 received.
- 1 6. The apparatus of claim 5 wherein the control layer is further configured to
2 signal detachment of the wireless peripheral on the downstream port when
3 the retry count is exhausted.

1 7. The apparatus of claim 6 wherein the control layer is further configured to
2 mark the status register to indicate if any new wireless peripheral devices
3 have attached.

1 8. The apparatus of claim 7, the control layer further comprising:
2 a protocol management interface configured to interface using Universal Serial
3 Bus protocol.

1 9. A method of attaching wireless peripheral devices to a wireless host
2 device, the method comprising:
3 sending a polling message to the downstream port;
4 receiving a response from the wireless peripheral device on the downstream port;
5 determining the state of the downstream port based on the response;
6 updating port information with the state of the downstream port; and
7 updating a status register regarding the state of the downstream port.

1 10. The method of claim 9 wherein the polling message comprises a message
2 header identifier.

1 11. The method of claim 10 further comprising:
2 re-polling the downstream port when a response is not received from the wireless
3 peripheral device on the downstream port.

1 12. The method of claim 11 wherein the re-polling is done until a retry count
2 is exhausted.

1 13. The method of claim 12 further comprising:
2 signaling detachment of the wireless peripheral on the downstream port when the
3 retry count is exhausted.

1 14. The method of claim 13 further comprising:

2 marking the status register to indicate if any new wireless peripheral devices have
3 attached.

1 15. A computer-readable medium encoded with data and instructions, the data
2 and instructions causing an apparatus executing the instructions to:

3 poll a downstream port;

4 receive a response from the wireless peripheral device on the downstream port;

5 determine the state of the downstream port based on the response;

6 update port information with the state of the downstream port; and

7 update a status register regarding the state of the downstream port.

1 16. The computer-readable medium of claim 15 wherein the polling of the
2 downstream port is accomplished by sending a polling message, the
3 polling message comprising a message header identifier.

1 17. The computer-readable medium of claim 16, the instructions further
2 causing an apparatus executing the instructions to:

3 re-poll the downstream port when a response is not received from the wireless
4 peripheral device on the downstream port.

1 18. The computer-readable medium of claim 17 wherein the re-polling is done
2 until a retry count is exhausted.

1 19. The computer-readable medium of claim 18, the instructions further
2 causing an apparatus executing the instructions to:

3 signal detachment of the wireless peripheral on the downstream port when the
4 retry count is exhausted.

1 20. The computer-readable medium of claim 19, the instructions further
2 causing an apparatus executing the instructions to:

3 mark the status register to indicate if any new wireless peripheral devices have
4 attached.

1 21. An apparatus comprising:
2 means for polling a downstream port;
3 means for receiving a response from the wireless peripheral device on the
4 downstream port;
5 means for determining the state of the downstream port based on the response;
6 means for updating port information with the state of the downstream port; and
7 means for updating a status register regarding the state of the downstream port.

1 22. The apparatus of claim 21 wherein the means for polling of the
2 downstream port is configured to send a polling message, the polling
3 message comprising a message header identifier.

1 23. The apparatus of claim 22 further comprising:
2 means for re-polling the downstream port when a response is not received from
3 the wireless peripheral device on the downstream port.

1 24. The apparatus of claim 23 wherein the downstream port is re-pollled until a
2 retry count is exhausted.

1 25. The apparatus of claim 24 further comprising:
2 means for signaling detachment of the wireless peripheral on the downstream port
3 when the retry count is exhausted.

1 26. The apparatus of claim 25 further comprising:
2 means for marking a status register to indicate if any new wireless peripheral
3 devices have attached.

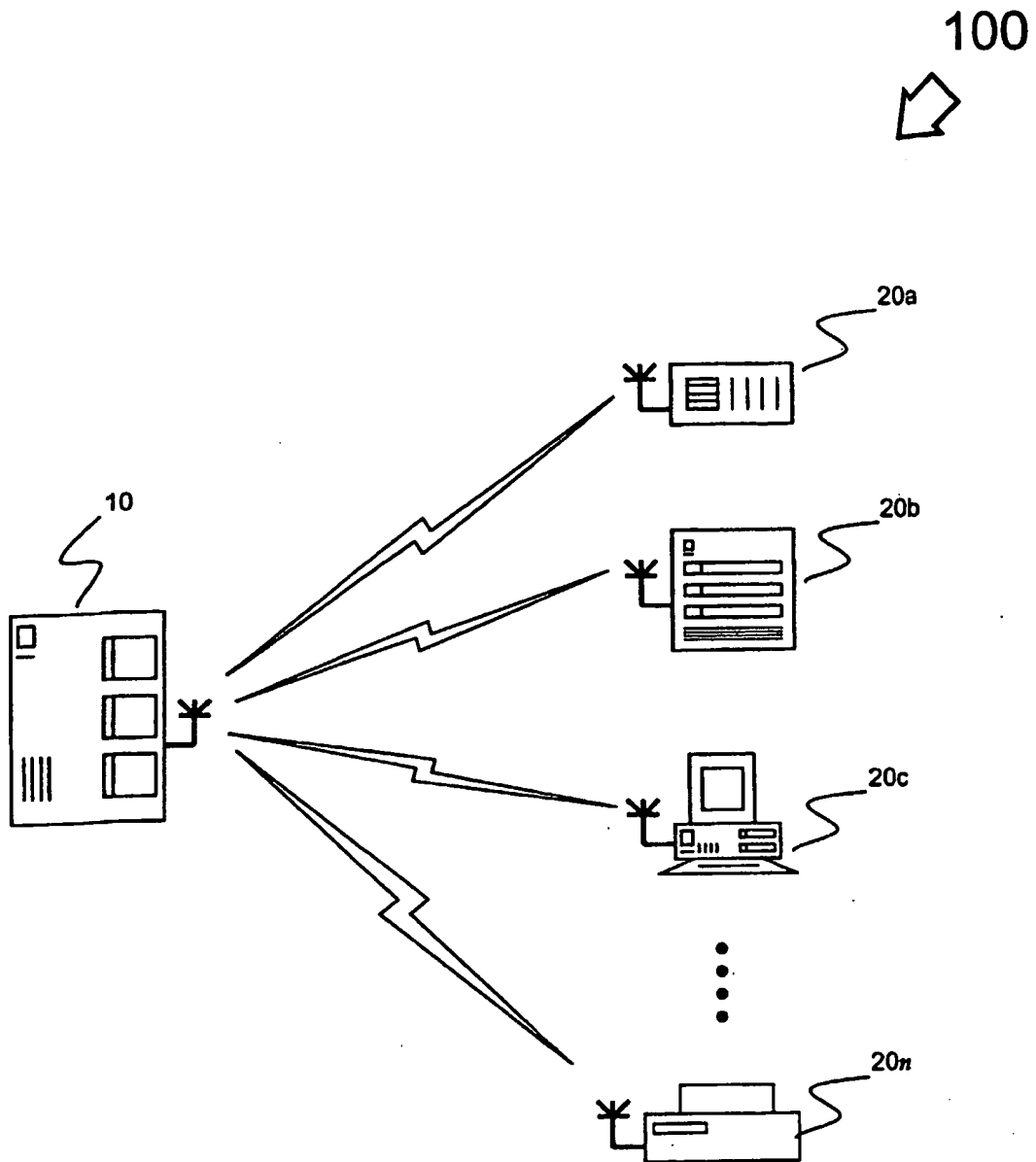


FIG. 1A

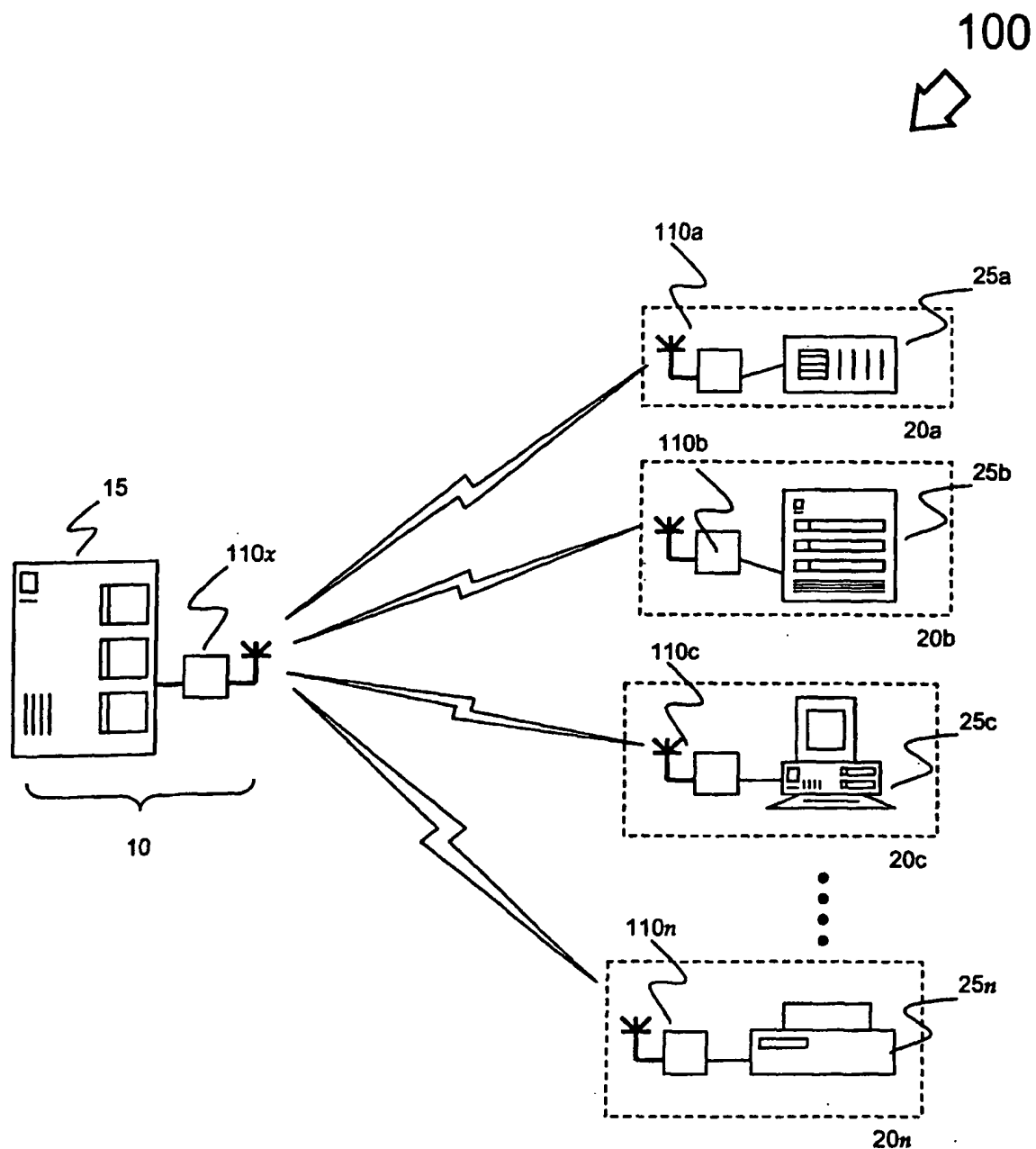


FIG. 1B

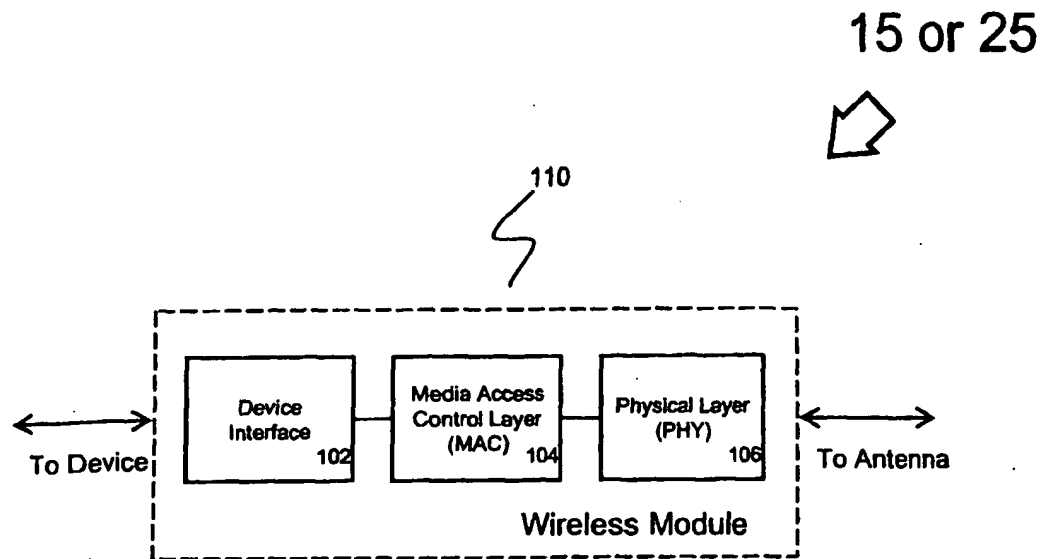


FIG. 2

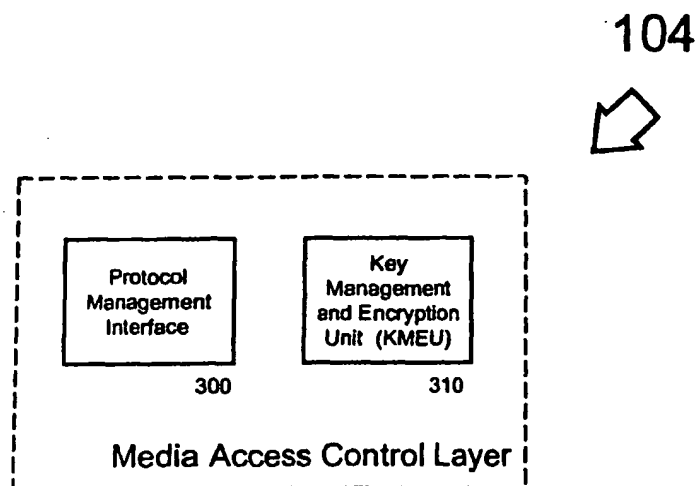


FIG. 3

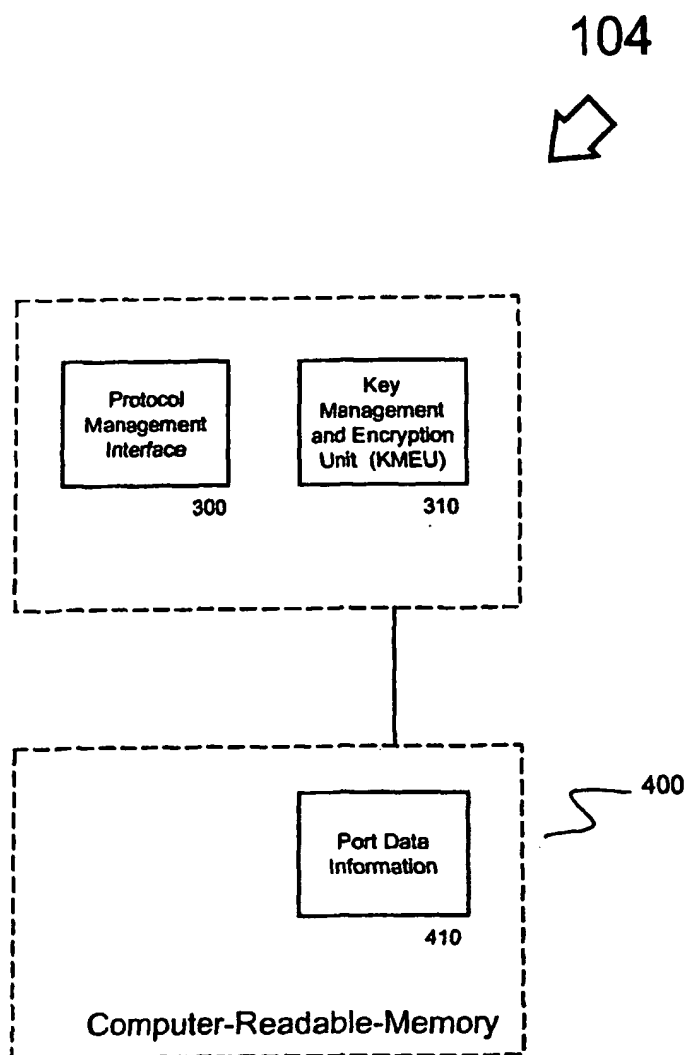


FIG. 4

500

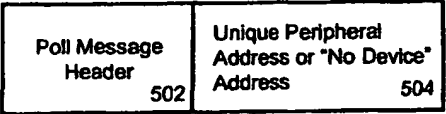


FIG. 5

600

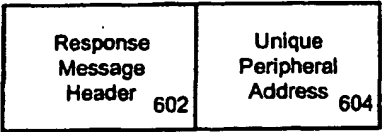


FIG. 6

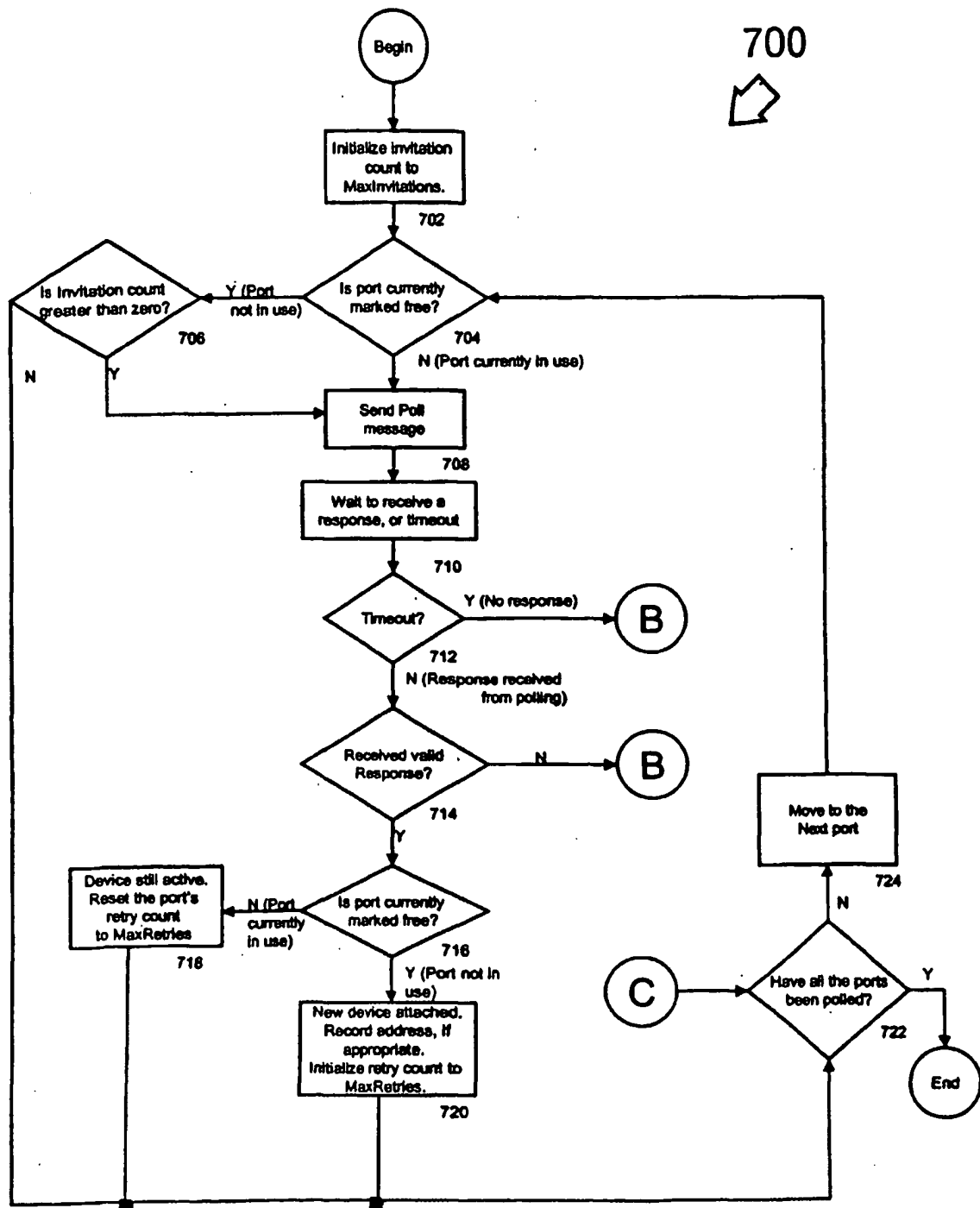


FIG. 7A

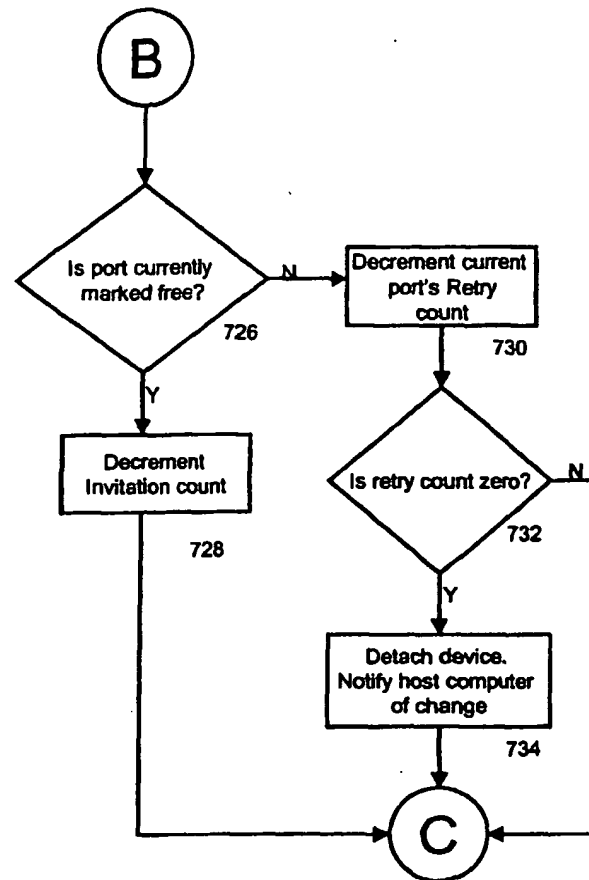


FIG. 7B

800

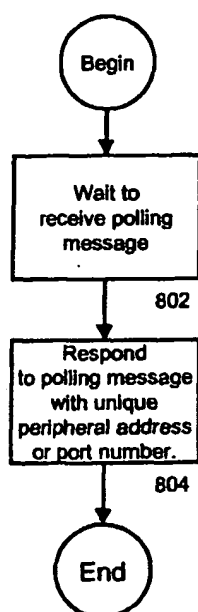


FIG. 8

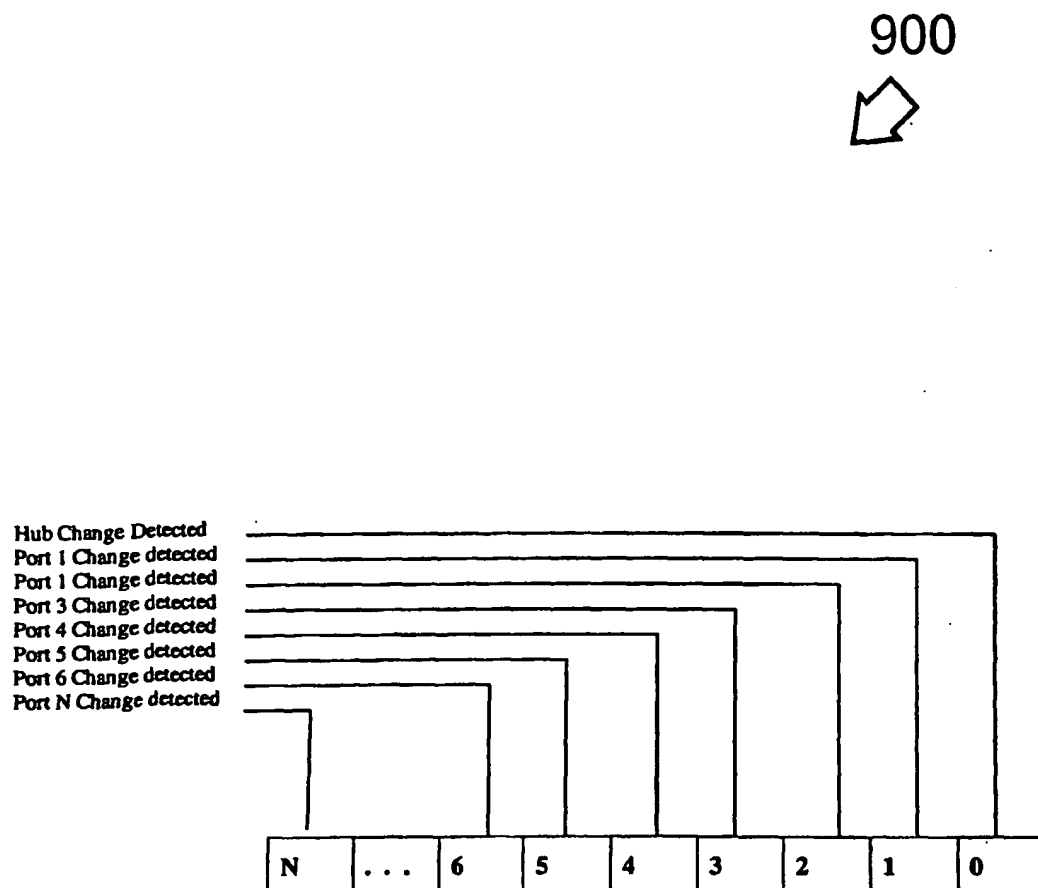


FIG. 9

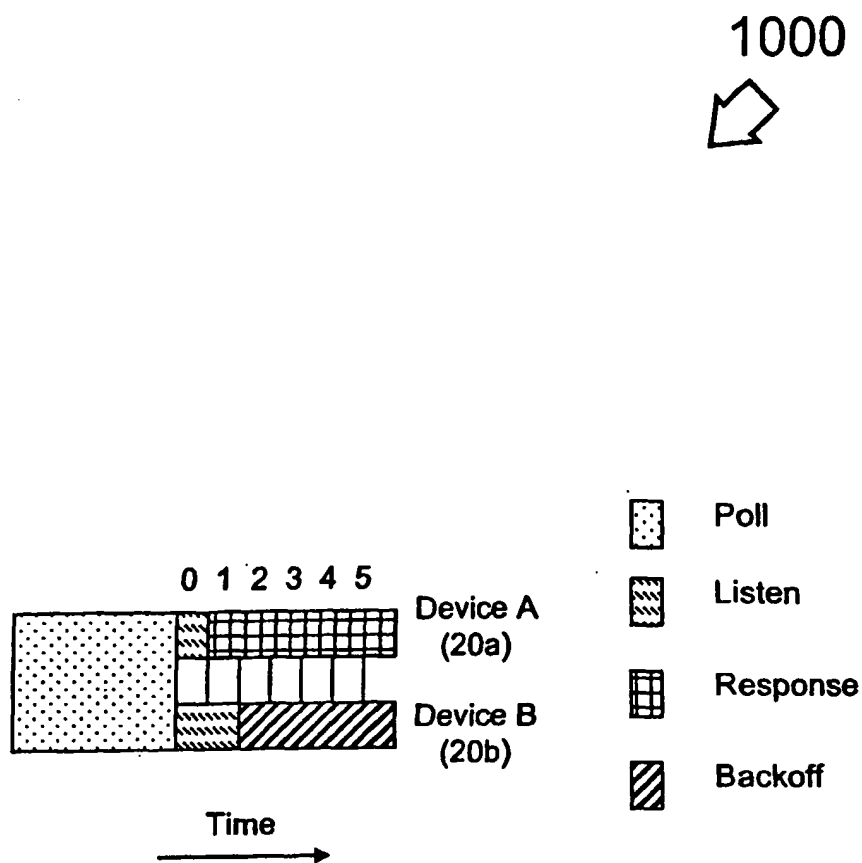


FIG. 10

Current State	Current Addr	Poll	Response	Retries	Host Action	New State
Detached	NoDevice	Poll (NoDevice)	-		-	Detached
Detached	NoDevice	Poll (NoDevice)	Resp (x)		Register & Enumerate	Attached
Attached	x	Poll (x)	Resp (x)		-	Attached
Attached	x	Poll (x)	-	> 0	Decrement Retries	Attached
Attached	x	Poll (x)	-	= 0	Deregister & Reset Retries	Detached

FIG. 11a

Port State	Poll Message	Device Attached		Device Detached	
		Resp Message	Host Device Action	Resp Message	Host Device Action
Detached	Poll /No Dev ID	Resp / ID	Set to ID	None	None
Attached	Poll / ID	Resp / ID	None	None	Set to Dev ID

FIG. 11b

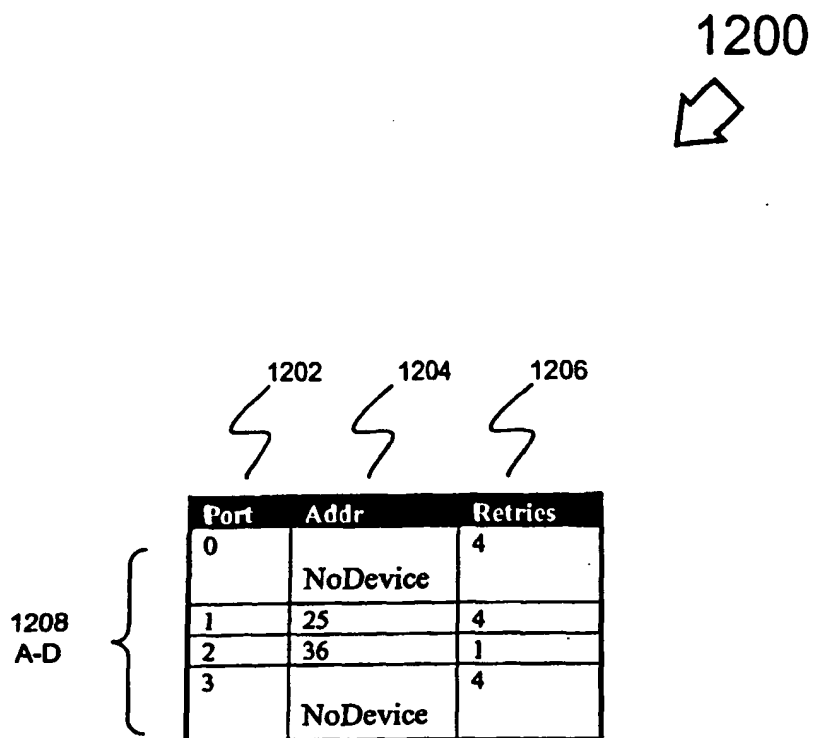
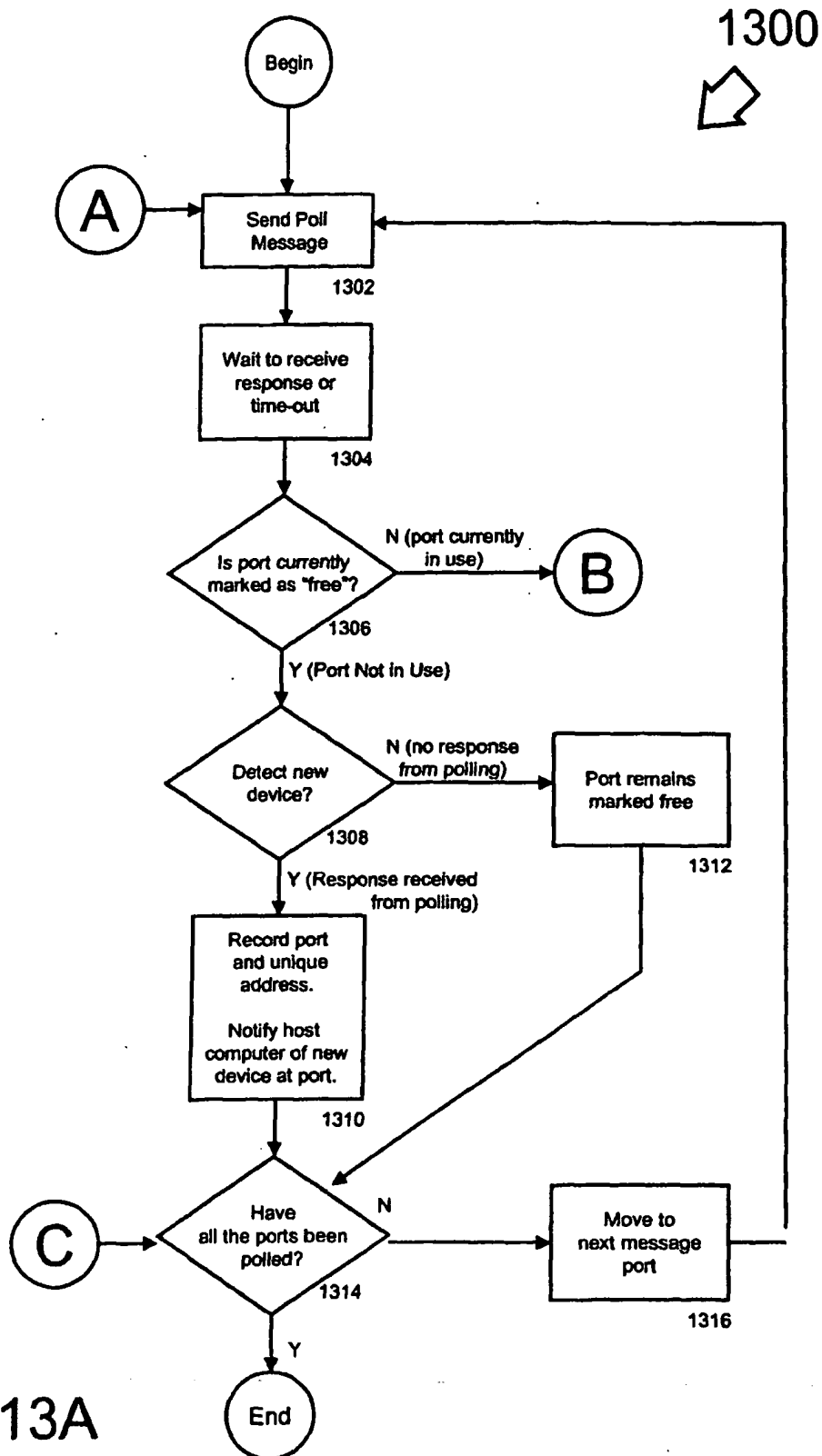


FIG. 12



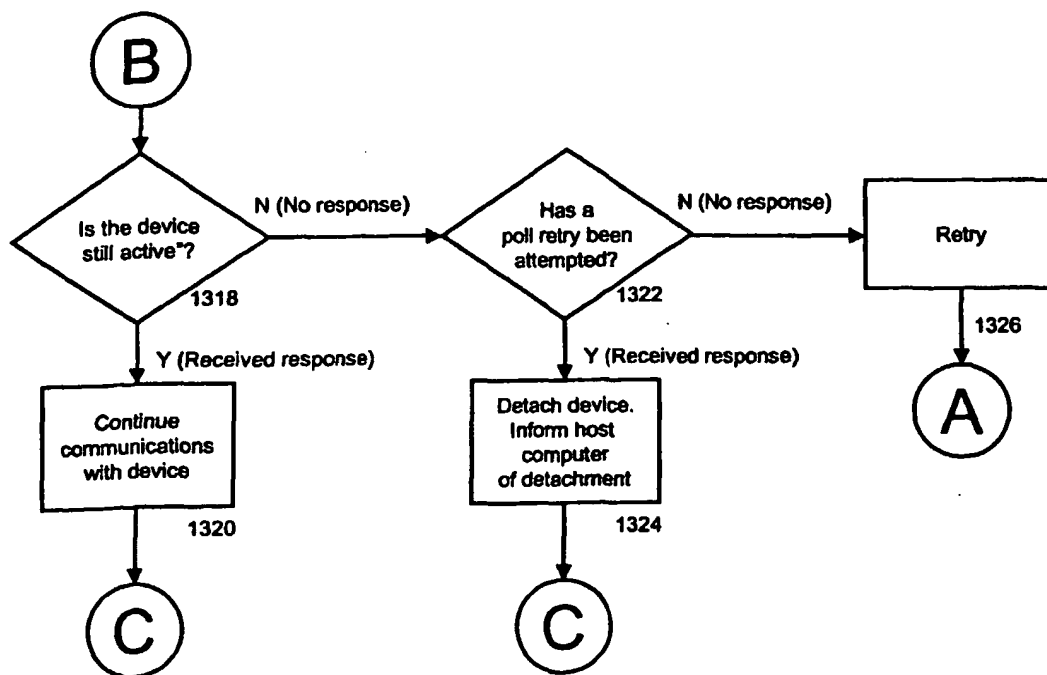


FIG. 13B

1400

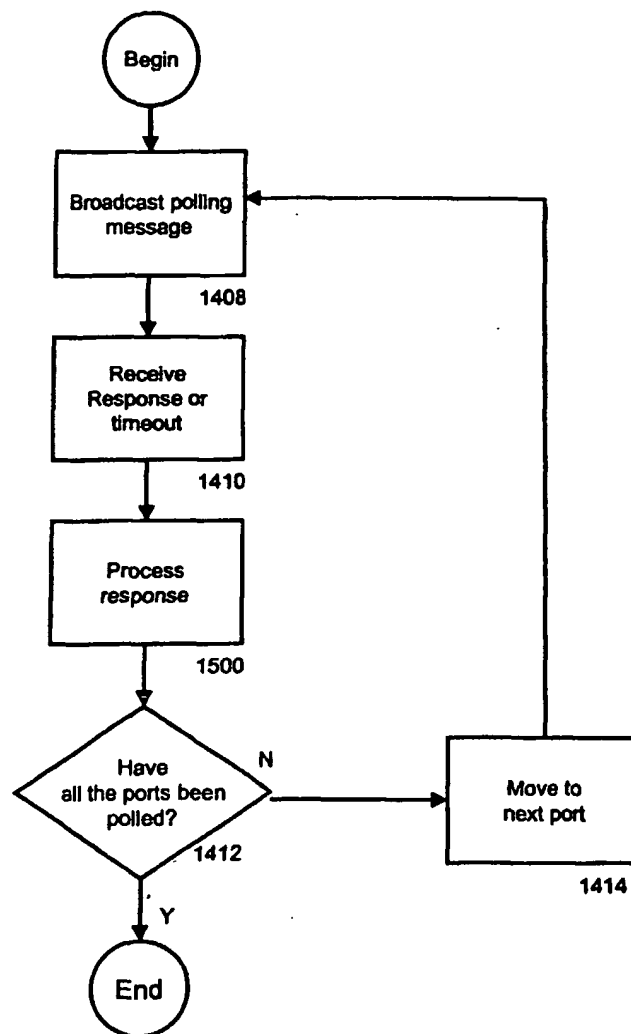


FIG. 14

1500

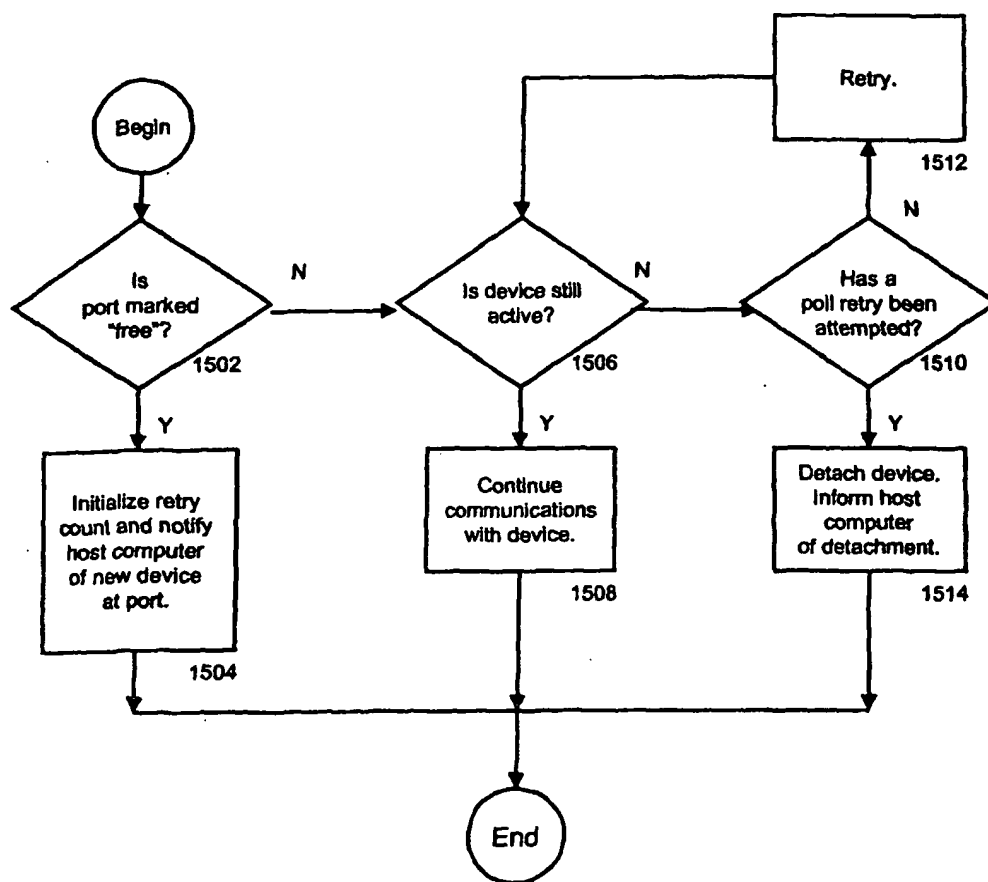


FIG. 15

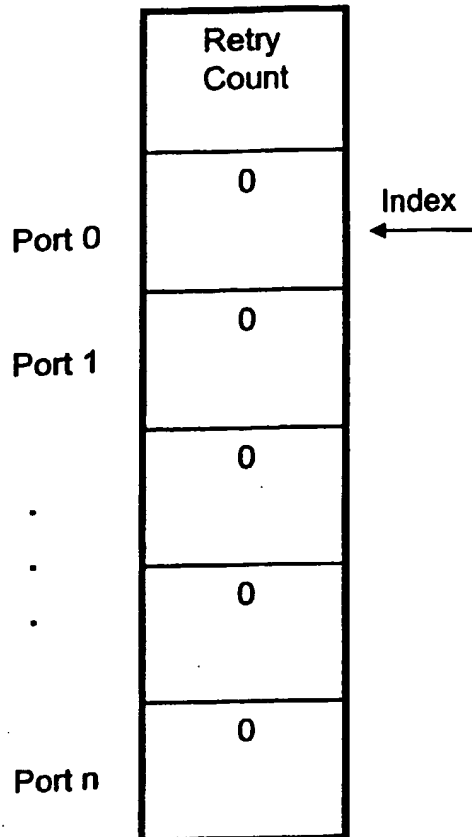


FIG. 16A

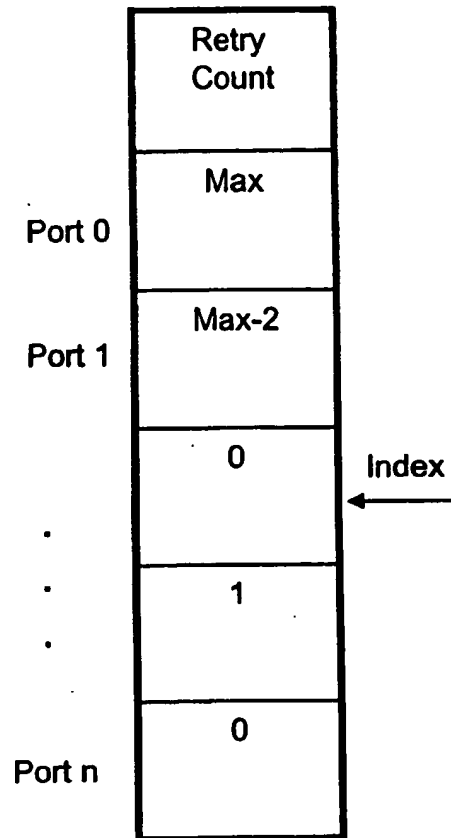


FIG. 16B

INTERNATIONAL SEARCH REPORT

International Application No.

PCT/US 02/26914

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F13/38

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, WPI Data, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	WO 01 56179 A (JONAS AMNON ; SHAHAR MENASHE (IL); VYYO LTD (US)) 2 August 2001 (2001-08-02) page 2, line 21 -page 3, line 19 page 5, line 5 - line 20	1-26
A	WO 01 11476 A (MILAN HENRY) 15 February 2001 (2001-02-15) page 4, line 2 -page 6, line 30 page 11, line 6 -page 12, line 4 abstract; figure 2	1-26
A	FR 2 799 288 A (IBM) 6 April 2001 (2001-04-06) page 3, line 23 -page 4, line 28 page 7, line 22 -page 10, line 6 abstract	1-26



Further documents are listed in the continuation of box C.



Patent family members are listed in annex.

* Special categories of cited documents :

- *A* document defining the general state of the art which is not considered to be of particular relevance
- *E* earlier document but published on or after the international filing date
- *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- *O* document referring to an oral disclosure, use, exhibition or other means
- *P* document published prior to the international filing date but later than the priority date claimed

- *T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- *X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- *Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
- *Z* document member of the same patent family

Date of the actual completion of the international search

20 December 2002

Date of mailing of the international search report

08/01/2003

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

Nguyen Xuan Hiep, C

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/US 02/26914

Patent document cited in search report		Publication date	Patent family member(s)	Publication date
WO 0156179	A	02-08-2001	AU 3798401 A	07-08-2001
			AU 3798501 A	07-08-2001
			WO 0156179 A1	02-08-2001
			WO 0156180 A1	02-08-2001
			US 2002036985 A1	28-03-2002
			US 2002115421 A1	22-08-2002
			AU 3799801 A	07-08-2001
			WO 0156200 A1	02-08-2001
			US 2002009970 A1	24-01-2002
WO 0111476	A	15-02-2001	AU 6517000 A	05-03-2001
			WO 0111476 A1	15-02-2001
FR 2799288	A	06-04-2001	FR 2799288 A1	06-04-2001

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☒ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.